8. Polina

Program Name: Polina.java Input File: polina.dat

In a recent CS lesson on encryption, the students were given an assignment to work with a partner to create their own encryption algorithm for encoding and decoding messages. After much thought, trial and error, Polina and her partner decided on the following encoding process.

1. For each code number following the message, convert it to an equivalent three-digit hexadecimal value. (They were careful to choose code numbers that would always convert to three digits in hex.)
2. If the first digit of the code number was even, for step 3 they would start counting from the left of the current message, otherwise they would count from the right side.
3. Switch the message characters in the places given by the 2nd and 3rd digits of the code number, counting from the side determined by step 2. (Polina was also careful to make sure these two digits for each code number were different, and also designated positions within the length of the message.)
4. Add an "A" to the left side of the message and "AA"" to the right side.
5. Add the 1st character of the hex value to the left side of the message and the 3rd character to the right side.
6. Repeat steps 1, 2, 3 and 5 for each additional code number.

For example, if the message was "INVITATIONAL" and the code number is 914 (392 in hex), the encoding result would be "3AINVITATILNAOAA2". Since the code number first digit is odd, they counted from the right 1 and 4 places, and switching the characters "L" and "O", which resulted in "INVITATILNAO". They then added an "A" to the front and "AA" to the back, and finally put the first hex digit (3) at the front and the last hex digit (2) at the back, resulting in the final encoded message, "3AINVITATILNAOAA2".

For the message "DISTRICT" followed by two code numbers, 425 (1A9) and 562 (232), the first encoded message was "1ADRSTIICTAA9", and the final one "21ADRSTIACTAI92".

Input: Several messages, each on one line, followed by one or more code numbers (no more than ten) to be used to encode the message. All messages are single words containing all uppercase letters and no symbols or spaces.

Output: The encoded message according to the rules stipulated above, with all letters in uppercase.

Sample input:

```
INVITATIONAL 914
DISTRICT 425 562
UILCONTEST 691 472 456
```

Sample output:

```
INVITATIONAL 3AINVITATILNAOAA2
DISTRICT 21ADRSTIACTAI92
UILCONTEST 112OSLICANTEUTAA388
```